15th European Signal Processing Conference (EUSIPCO 2007), Poznan, Poland, September 3-7, 2007, copyright by EURASIP

A FRAMEWORK FOR ANALYSIS OF MUSIC SIMILARITY MEASURES

Jesper Højvang Jensen, Mads Græsbøll Christensen, and Søren Holdt Jensen

Department of Electronic Systems
Aalborg University, Denmark
email: {jhj, mgc, shj}@es.aau.dk

ABSTRACT

To analyze specific properties of music similarity measures that the commonly used genre classification evaluation procedure does not reveal, we introduce a MIDI based test framework for music similarity measures. We introduce the framework by example and thus outline an experiment to analyze the dependency of a music similarity measure on the instrumentation of a song compared to the melody, and to analyze its sensitivity to transpositions.

Using the outlined experiment, we analyze music similarity measures from three software packages, namely Marsyas, MA toolbox and Intelligent Sound Processing toolbox. The tested timbral similarity measures perform instrument recognition relatively well, although they are sensitive to transpositions and differences between sound fonts. The beat/rhythm/melody similarity measures are not always able to recognize the same melody played with different instruments.

1. INTRODUCTION

As sound compression has matured and storage has become cheap, digital music collections, e.g. in the mp3 format, have grown very large. Navigation in such collections is limited by the metadata, such as title and artist, that is associated with the songs. Appropriate music similarity measures could help navigating in such collections, and could also be used for music recommendation systems in online music stores. By music similarity measure, we mean a quantitative measure of the similarity (or distance) between some musical aspects of two songs. Most music similarity measures are divided into a feature extraction part that extracts features that compactly describe some musical aspects of a song, and a distance measure that computes the distance between songs from the features. Much work has already been done on music similarity and on the related task of genre classification, e.g. [1–9]. Genre classification is often used to evaluate music similarity measures since it simplifies evaluation compared to the numerous user evaluations that are otherwise needed.

While genre classification provides a good first estimate of the performance of a music similarity measure, it does not provide details of its inner workings. For example, a commonly used feature that performs relatively well in genre classification when combined with a classifier, is the mel-frequency cepstral coefficients (MFCCs), e.g. [3–8]. The MFCCs have their origins in speech recognition, where they are used to model the spectral envelope of a single speaker while suppressing the fundamental frequency. This is consistent with the common notion in music similarity of MFCCs as a timbral descriptor. Timbre is defined as "the auditory sensation in terms of which a listener can judge that two sounds with the same loudness and pitch are dissimilar" [10]. Since the temporal envelope and the spectral envelope play key roles to the perception of timbre [11], one would expect the MFCCs to mainly depend on the instrumentation of a song, and one would expect them to perform genre classification by matching songs with similar instrumentation. It is a tempting conclusion, but there are a number of uncertainties. For instance, in music several notes are often played at once, and it is not obvious how this mixing affects the spectral envelope. Furthermore, it is well-known that MFCCs are not completely independent of the fundamental frequency (e.g. [12]). Unfortunately, the good performance in genre classification does not reveal to which extent the MFCCs reflect the instrumentation, and to which extent they reflect the harmonies and key of a song. In this paper, we take the first steps towards an answer by introducing a test framework based on MIDI synthesis that supplements the genre classification results.

In Section 2, we describe an experiment to evaluate the dependency of a similarity measure on instrumentation compared to melody and an experiment to evaluate the sensitivity of a similarity measure to transpositions. In Section 3 and Section 4, we briefly describe some similarity measures we have evaluated using the experimental setups and present the results, respectively. In Section 5, we discuss the results as well as how the experiments can be modified to analyze other aspects of music similarity measures.

2. ANALYSIS FRAMEWORK

To analyze how a music similarity measure depends on the instrumentation compared to the notes and how it is affected by transpositions, we use a MIDI file[1] setup. We take a number of MIDI files, manipulate the instrumentation and key using the MATLAB MIDI-toolbox [13], and then use a software synthesizer to generate waveform signals that can be used in a nearest neighbor classifier. Using MIDI files might bias the results, since the synthesized signal will be more homogeneous than recordings of real musicians would be. The advantage is that it allows us to manipulate instrumentation, tempo and melody in a flexible, reproducible way. In what follows, we first introduce an experiment to test the dependency of a musical similarity measure on instrumentation compared to the dependency on the notes, i.e., the melody and harmonies. Second, we introduce an experiment to eval-

This research was supported by the Intelligent Sound project, Danish Technical Research Council grant no. 26–04–0092, and the Parametric Audio Processing project, Danish Research Council for Technology and Production Sciences grant no. 274–06–0521.

[1] A MIDI file contains information about fundamental frequency, instrumentation and onset/duration of all notes.

uate the dependency on transpositions of a song, i.e., how shifting all notes of a song a number of semitones affects the similarity measure.

2.1 Instrumentation versus notes

The dependency on instrumentation is tested by taking M different MIDI songs and choosing N instruments. We choose songs of different musical styles to ensure that the songs will have very different harmonic and melodic characteristics. For each m, from 1 to M, and for each n, from 1 to N, we do the following:

1. Read MIDI file m.
2. Remove all percussive instruments.
3. Let all notes be played by instrument n.
4. Synthesize a waveform signal.
5. Extract the feature vector $\mathbf{v}_{mn}$ from the waveform signal.

In the following, we will use the term "melody" to denote the instrument and key-independent part of a song, i.e., we will say that all songs created from MIDI file m, which share the exact same melody, harmonies and timing, share the same melody no matter what the instrumentation or key is. After computing features for all $M \times N$ songs, we find the nearest neighbor of $\mathbf{v}_{mn}$ according to the distance $d(\cdot,\cdot)$ associated with the feature:

$$(p,q) = \underset{\substack{l,k \\ (l,k)\neq(m,n)}}{\arg\min}\, d(\mathbf{v}_{mn}, \mathbf{v}_{lk}) \tag{1}$$

Let $\mathbf{v}_{mn}$ be a given query, and let the nearest neighbor among the target songs be $\mathbf{v}_{pq}$. If $p = m$, then the nearest neighbor has the same melody as the query. We define the melody classification accuracy by the fraction of the $M \times N$ queries where the nearest neighbor has the same melody. Similarly, we define the instrument classification accuracy as the fraction of queries where the nearest neighbor uses the same instrument.

2.2 Transpositions

A human listener does not consider the same song played in different keys as different songs. Similarly, an instrument playing two different notes is still considered the same instrument. For most similarity measures it is therefore of interest to know how sensitive they are to transpositions. This is what this experiment investigates. It is similar to the previous experiment; the differences being that the tonal range is normalized and the song is transposed. The tonal range is normalized by transposing each individual track of a song (such as bass or melody) by an integer number of octaves, such that the average note being played in a track is as close as possible to the C4 note (middle C on the piano). The constraint of transposing tracks an integer number of octaves ensures that the harmonic relationships are not changed. As before, let m and n denote melody and instrument number, and let s denote the number of semitones a song is transposed. Features $\mathbf{v}_{mn}^{(s)}$ are computed for different values of s. When evaluating (1), a query that has not been transposed is always used, i.e. the minimization is over $d(\mathbf{v}_{mn}^{(0)}, \mathbf{v}_{lk}^{(s)})$. Melody and instrument classification rates are computed for all values of s.

2.3 Implementation of the framework

In genre classification, standard data sets such as the training data from the ISMIR 2004 Genre Classification contest [14] is readily available. However, for our purpose there is no obvious MIDI data set to use. For this reason we created 112 songs of length 30 s using Microsoft Music Producer, a program for automatically creating MIDI files for background music. Each song has a different musical style with different melody, rhythm, tempo, accompaniment and instrumentation. Examples of styles are "50s rock", "Latin" and "Southern rock". From the General MIDI Level 1 Sound Set, all the 112 instruments that neither belong to the percussive instrument family nor are sound effects (see [15]), were selected. Of the 112 instruments and 112 songs, ten random subsets of 30 songs and 30 instruments were chosen. For each subset, the experiments described in Section 2.1 and 2.2 were performed. To synthesize waveform signals from MIDI, the software synthesizer TiMidity++ was used. Two different sound fonts, Fluid R3 and SGM-180 v1.5 GM, were used. Equation (1) was evaluated both with query and target synthesized from the same sound fonts, and with query and target synthesized from different sound fonts. All feature extraction routines where given a mono signal sampled at 22 kHz as input.

3. MUSIC SIMILARITY MEASURES

In this section, the tested similarity measures are described. Music similarity measures from three different publicly available software packages have been tested: Marsyas [16], the MA toolbox [17], and the Intelligent Sound Processing toolbox (see `http://isound.kom.auc.dk/`). Since not all of the similarity measures incorporate a distance measure between individual songs, some ad hoc distance measures have been introduced. These are also described below.

3.1 Marsyas

From Marsyas v. 0.1, five feature sets are tested. The feature sets are thoroughly described in [3], where they were used with a probabilistic classifier that was trained on features from an entire genre. For this reason, a distance measure between feature vectors from individual songs does not exist. For all but the *beat* feature, which performed better with ordinary Euclidean distance, we therefore use the weighted Euclidean distance, $d_{\boldsymbol{W}}(\mathbf{u},\mathbf{v}) = (\mathbf{u}-\mathbf{v})^{\mathrm{T}}\boldsymbol{W}(\mathbf{u}-\mathbf{v})$, where $\boldsymbol{W}$ is a diagonal matrix with positive elements. For the timbre features, $\boldsymbol{W}$ was chosen to maximize the difference between the average distance between all vectors and the average distance between vectors from songs with the same instrument, subject to $\|\boldsymbol{W}\|_{\mathrm{F}} = 1$, where $\|\cdot\|_{\mathrm{F}}$ is the Frobenius norm:

$$\boldsymbol{W} = \underset{\substack{\boldsymbol{W} \\ \|\boldsymbol{W}\|_{\mathrm{F}}=1}}{\arg\max} \left[\frac{1}{M^2N^2} \sum_{i,j} \sum_{k,l} d_{\boldsymbol{W}}(\mathbf{v}_{ij}, \mathbf{v}_{kl}) - \frac{1}{M^2N} \sum_{j} \sum_{i,k} d_{\boldsymbol{W}}(\mathbf{v}_{ij}, \mathbf{v}_{kj}) \right]. \tag{2}$$

Before computing $\boldsymbol{W}$, all feature dimensions were normalized to have unit variance. For the pitch feature, the average distance between songs of the same melody was minimized instead. The weights $\boldsymbol{W}$ were computed from one of the 30×30 subsets from the experiment in Section 2.1 where both query

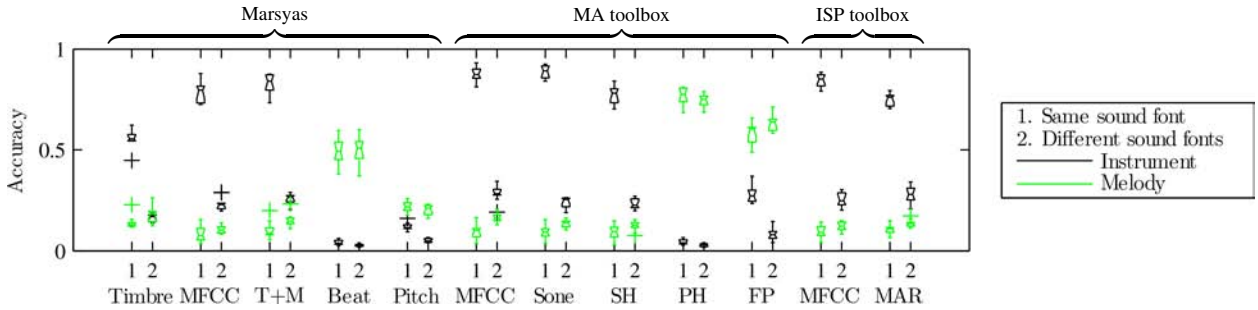

Figure 1: Instrument and melody classification accuracies. The number 1 denotes that the same sound font has been used for both query and target, while 2 denote that different sound fonts were used for the query and target. The whiskers on the plot denote the 95% confidence intervals.

and target songs were synthesized with the Fluid R3 sound font. The same weights were used for the transposition experiment. In the following, the five feature sets from Marsyas we have tested are described:

Timbre: Mean and variance of the spectral centroid, roll-off, flux and of the fraction of low-energy frames [3]. Distance measure: Weighted Euclidean.

MFCC: Mean and variance of the first five mel-frequency cepstral coefficients (MFCCs) [3]. Distance measure: Weighted Euclidean.

Timbre + MFCC (T+M): Concatenation of the timbre and MFCC features [3]. Distance measure: Weighted Euclidean.

Beat: Based on a histogram of prominent beats. Consists of the amplitudes and periods of the two first peaks in the histogram, the ratio between these two peaks, and the sum of all peaks [3]. Distance measure: Euclidean.

Pitch: Derived from a histogram of pitches in the signal. Contains among others periods and amplitudes of some of the most prominent peaks on both a full semitone scale and on an octave-independent (modulus 12) scale [3]. Distance measure: Weighted Euclidean.

3.2 MA toolbox

From the MA toolbox [17], five features were tested. The distance measures recommended in [17] are used.

MFCC: MFCCs are estimated in short windows, and a Gaussian mixture model (GMM) is trained to model them. Distance measure: Approximated, symmetrized Kullback-Leibler [7].

Sone: In a number of frequency bands distributed according to the Bark scale, the loudness measured in sone is computed. A GMM is trained on the loudness values. Distance measure: Approximated, symmetrized Kullback-Leibler [7].

Spectrum histogram (SH): A derivative of the raw sone features where the number of times each loudness level has been exceeded in each frequency band is counted [17]. Distance measure: Euclidean.

Periodicity histogram (PH): A description of periodic beats [17]. Distance measure: Euclidean.

Fluctuation pattern (FP): Another approach to describe periodicities in a signal. Distance measure: Euclidean.

3.3 Intelligent Sound Processing toolbox

Two similarity measures from the Intelligent Sound Processing (ISP) toolbox were tested:

MFCC: Similar to the MA toolbox *MFCC*, but with different parameters, such as the number of dimensions. Distance measure: Approximated, symmetrized Kullback-Leibler.

MAR: A multivariate autoregressive model that captures temporal correlation of MFCCs over 1 s segments [18]. A feature vector is produced for every 1 s of audio. Distance measure: For each vector in the query song, the other songs are ranked according to their minimum distance to that vector. The average ranking is then used as the distance measure.

4. RESULTS

The results of the two experiments are plotted in Figures 1 and 2. As is seen, some of the results are highly dependent on whether the query features are synthesized from the same sound font as the target features or not. However, the results are largely independent of which of the two sound fonts is used as query and which is used as target. Therefore, only results for Fluid 3 as both query and target, and results for Fluid 3 as query and SGM 180 as target are shown.

When query and target are from the same sound font, the timbral similarity measures perform well. The Marsyas *Timbre+MFCC*, the MA toolbox *MFCC* and *Sone*, and the ISP toolbox *MFCC* all have average instrument classification ratios in the range from 83% to 92%. The Marsyas *MFCC*, MA toolbox *spectrum histogram* and ISP toolbox *MFCC-MAR* also have relatively good performance, ranging from 75% to 79%. The Marsyas *timbre* performs worst of the timbral features with 55%. However, when query and target are from different sound fonts, the average instrument classification accuracy never exceeds 30%. Since the difference between the same instrument synthesized with different sound fonts is clearly audible, this is understandable, although still undesirable. According to [19], temporal characteristics such as attack transients contribute significantly to human perception of timbre. Timbre similarity measures that better incorporate this short-time temporal development might be less sensitive to the use of different sound fonts.

With respect to melody classification, three similarity measures are noteworthy: Marsyas *beat*, and MA toolbox *periodicity histogram* and *fluctuation pattern* with average classification accuracies of 51%, 78% and 62%, respectively. They are all practically independent of the combination of sound fonts used. The Marsyas *pitch* feature performs surprisingly bad, probably due to the inherently difficult problem of estimating multiple fundamental frequencies. Interestingly, the *fluctuation pattern* from the MA toolbox also performs better than random for instrument classification. Since in this laboratory setup neither the melody, rhythm, ac-

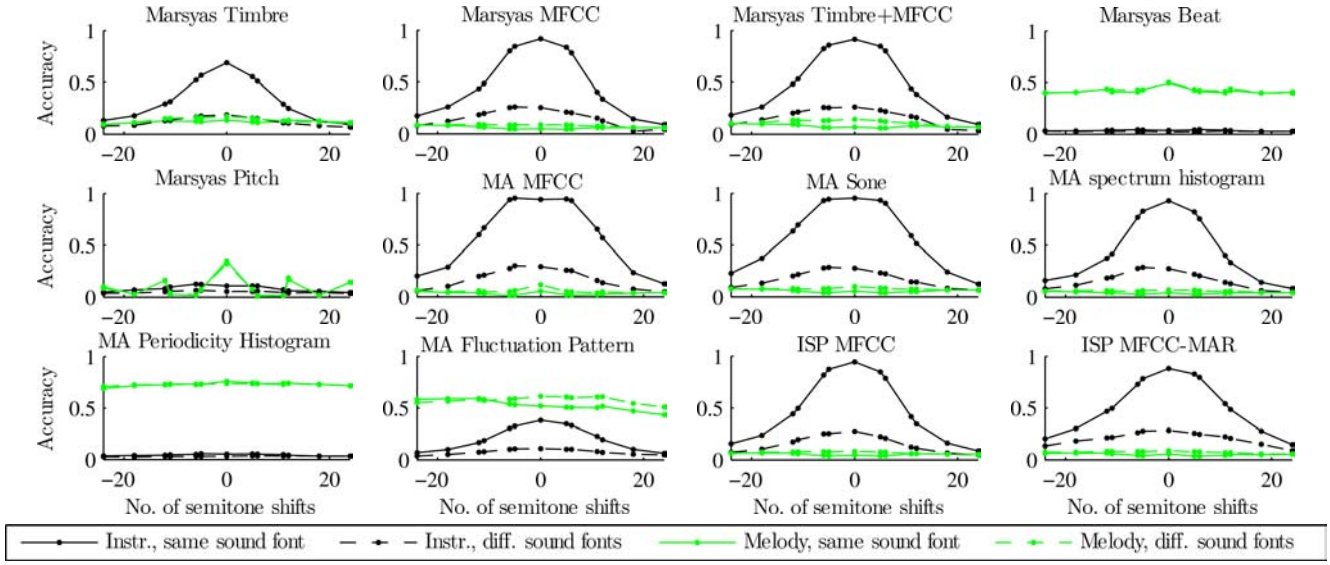

Figure 2: Sensitivity of music similarity measures to transpositions.

companiment nor timing changes, it ought to be possible to classify all melodies correctly. We therefore see much room for improvement.

The second experiment shows that all the timbral similarity measures behave similarly when exposed to transpositions. Accuracy is approximately halved when transposing 12 semitones (one octave). When transposing 24 semitones (two octaves), almost no instruments are recognized. An interesting detail is that the behavior of the *MFCC* feature from the MA toolbox is more similar to the *sone* feature from the same toolbox than it is to the *MFCC* feature from the ISP toolbox. The reason might be that the former two use the same statistical model. The features that performed well in melody recognition, namely MA toolbox *periodicity histogram* and *fluctuation pattern* and Marsyas *beat*, are all practically unaffected by transpositions. The Marsyas *pitch* feature is sensitive to transpositions, but since it contains information about the most dominant pitch, this is not surprising.

5. DISCUSSION

From the experiments we observed that the timbral similarity measures did not generalize well to different sound fonts. We therefore hypothesize that timbral similarity measures that also rely on the temporal envelope will better reflect the human sound perception where certain smooth spectral changes, such as adjusting the bass and treble, do not significantly alter the perception of timbre. We also observed that there is room for improvements with melody recognition.

These results could not have been obtained from a genre classification experiment alone. By using MIDI files, we have effectively separated the effect of instrumentation and melody, and a signal modification that would have been difficult or cumbersome to introduce directly in a waveform signal, namely transposition, has been introduced.

Although in this paper we have only tested the sensitivity of similarity measures to transpositions, it would also be relevant to measure the dependency on tempo, combinations of instruments, bandwidth and audio compression. We strongly recommend the use of such tests as a simple, yet insightful supplement to genre classification.

REFERENCES

[1] J. T. Foote, "Content-based retrieval of music and audio," in *Multimedia Storage and Archiving Systems II, Proc. of SPIE*, 1997, pp. 138–147.

[2] J.-J. Aucouturier and F. Pachet, "Finding songs that sound the same," in *Proc. of IEEE Benelux Workshop on Model based Processing and Coding of Audio*, 2002.

[3] G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," *IEEE Trans. Speech Audio Processing*, vol. 10, pp. 293–301, 2002.

[4] A. Flexer, "Statistical evaluation of music information retrieval experiments," Institute of Medical Cybernetics and Artificial Intelligence, Medical University of Vienna, Tech. Rep., 2005.

[5] J.-J. Aucouturier and F. Pachet, "Improving timbre similarity: How high's the sky?" *Journal of Negative Results in Speech and Audio Sciences*, vol. 1, no. 1, 2004.

[6] B. Logan and A. Salomon, "A music similarity function based on signal analysis," in *Proc. IEEE Int. Conf. Multimedia Expo*, 2001, pp. 745 – 748.

[7] E. Pampalk, "Speeding up music similarity," in *2nd Annual Music Information Retrieval eXchange*, London, 2005.

[8] M. I. Mandel and D. P. Ellis, "Song-level features and support vector machines for music classification," in *Proc. Int. Symp. on Music Information Retrieval*, 2005.

[9] J. Bergstra, N. Casagrande, D. Erhan, D. Eck, and B. Kégl, "Aggregate features and AdaBoost for music classification," *Machine Learning*, vol. 65, no. 2–3, pp. 473–484, 2006.

[10] *Acoustical Terminology SI*, New York: American Standards Association Std., Rev. 1-1960, 1960.

[11] B. C. J. Moore, *An introduction to the Psychology of Hearing*, 5th ed. Elsevier Academic Press, 2004.

[12] B. Milner and X. Shao, "Prediction of fundamental frequency and voicing from mel-frequency cepstral coefficients for unconstrained speech reconstruction," *IEEE Trans. Audio, Speech, Language Processing*, vol. 15, no. 1, pp. 24–33, 2007.

[13] T. Eerola and P. Toiviainen, "MIDI toolbox: Matlab tools for music research," University of Jyväskylä: Kopijyvä, Jyväskylä, Finland, Tech. Rep., 2004.

[14] ISMIR 2004 audio description contest – genre/artist ID classification and artist similarity. [Online]. Available: \url{http://ismir2004.ismir.net/genre_contest/index.htm}

[15] *General MIDI Level 1 Sound Set*, MIDI Manufacturers Association, 2007. [Online]. Available: http://www.midi.org/about-midi/gm/gm1sound.shtml

[16] G. Tzanetakis and P. Cook, "Marsyas: A framework for audio analysis," *Organized Sound*, vol. 4, no. 3, 2000.

[17] E. Pampalk, "A Matlab toolbox to compute music similarity from audio," in *Proc. Int. Symp. on Music Information Retrieval*, 2004, pp. 254–257.

[18] A. Meng, P. Ahrendt, and J. Larsen, "Improving music genre classification by short-time feature integration," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, 2005, pp. 497–500.

[19] T. D. Rossing, F. R. Moore, and P. A. Wheeler, *The Science of Sound*, 3rd ed. Addison-Wesley, 2002.